Дискретная математика

Лямбда-исчисление

Константин Чухарев

Осень 2026

Идея

> *Цель абстракции — не расплывчатость, а новый уровень точности.*
>
> ***— Эдсгер Дейкстра***

Вычисление как подстановка

Пример

$(\lambda x.x)y \to y$.

Тождественная функция, применённая к y.

$$(\lambda x.\lambda y.x)ab \to a$$

— первый аргумент, второй игнорируется.

Вычисление — подстановка.

Никаких чисел, строк, циклов — только функции.

Три конструктора

Определение 1 (λ-термы)

- **Переменная**: x.
- **Абстракция**: $\lambda x.M$ — функция от x.
- **Аппликация**: MN — применить M к N.

Пример

- $(\lambda x.x)y$ — аппликация абстракции к переменной.
- $\lambda xy.x$ — вложенная абстракция.

Соглашения

Три правила — универсальный язык.

Этого достаточно для любого вычисления.

Соглашения:

- $MNP = (MN)P$ (лево-ассоциативно).
- $\lambda xy.M = \lambda x.(\lambda y.M)$.

Синтаксис и редукция

Свободные и связанные

Определение 2

- **Свободная** переменная — та, на пути от которой к корню терма нет связывающего её λ.
- **Связанная**: есть λx.
- **Комбинатор**: замкнутый терм, то есть терм без свободных переменных.

Пример

$I = \lambda x.x$ — комбинатор.

$\lambda x.xy$ — не комбинатор: y свободна.

Свободные переменные мы заменяем, а связанные — нет.

Как параметр и локальная переменная функции.

α-конверсия и захват

Определение 3 (α-конверсия)

Переименование связанной переменной не меняет смысл терма: $\lambda x.M \overset{\alpha}{=} \lambda y.M[x := y]$.

Условие: y не входит свободно в M.

Пример

$\lambda x.x = \lambda y.y$ — α-эквивалентны.

$\lambda x.y \neq \lambda y.y$: свободная y была бы захвачена.

$(\lambda y.xy)[x := y] \to \lambda z.yz$ (сначала y переименовали в z), а не $\lambda y.yy$.

Подстановка $M[x := N]$ всегда избегает захвата.

α-конверсия и захват [2]

Свободная переменная не должна становиться связанной.

β-редукция

Определение 4

$(\lambda x.M)N \overset{\beta}{\to} M[x := N]$.

Подставить N вместо x в тело, избегая захвата.

Пример

$(\lambda x.xx)y \to yy$.

$(\lambda x.\lambda y.x)ab \to (\lambda y.a)b \to a$.

Редекс — аппликация, где слева абстракция.

Редукция — один шаг вычисления.

Нормальная форма и Ω

Определение 5 (Нормальная форма)

Нормальная форма — это терм, в котором нет ни одного β-редекса.

M имеет нормальную форму N, если M редуцируется к N и в N нет редексов.

Пример

$(\lambda x.x)(\lambda y.y) \to \lambda y.y$ — нормальная форма.

$\Omega = (\lambda x.xx)(\lambda x.xx) \to \Omega$.

Единственный шаг воспроизводит сам терм.

Не всякий терм имеет нормальную форму.

Ω — λ-аналог бесконечного цикла.

Конфлюэнтность

Теорема 1 (Чёрч–Россер)

Если терм редуцируется двумя путями, оба сходятся к общему терму.

Пример

$(\lambda x.x)((\lambda y.y)z)$ редуцируется и внешним, и внутренним редексом. Оба пути сходятся к z.

Порядок редукции не влияет на результат.

Нормальная форма единственна.

Свойство алмаза (*diamond property*): $M \to N_1$, $M \to N_2$ влекут общий N_3.

Стратегии редукции

Пример

$(\lambda x.y)\Omega$.

- Нормальная стратегия (самый левый внешний редекс): $(\lambda x.y)\Omega \xrightarrow{\beta} y$ за один шаг.
- Аппликативная (самый левый внутренний): сначала редуцирует Ω и зацикливается.

Порядок не влияет на результат, но влияет на завершение.

Нормальная стратегия находит нормальную форму, если она есть (теорема стандартизации).

Нормальная стратегия — это вызов по имени (Haskell), а аппликативная — вызов по значению (ML).

Кодирование

Булевы Чёрча

Определение 6

- $\text{true} = \lambda xy.x$.
- $\text{false} = \lambda xy.y$.
- Условное выражение — аппликация: $\text{true}\ AB \to A$.

Пример

- $\text{not} = \lambda b.b\ \text{false}\ \text{true}$.
- $\text{and} = \lambda ab.ab\ \text{false}$.
- $\text{or} = \lambda ab.a\ \text{true}\ b$.

Булево значение само делает выбор.

Никакого специального `if`.

Числа Чёрча

Определение 7

$n = \lambda fx.f^n(x)$ — n-кратная итерация.

$0 = \lambda fx.x$, $1 = \lambda fx.fx$, ...

Число Чёрча n формально есть $\lambda fx.f^n(x)$: итерация функции f ровно n раз.

- succ $= \lambda nfx.f(nfx)$.
- add $23 \xrightarrow[\beta]{} 5$.
- mult $23 \xrightarrow[\beta]{} 6$.
- exp $= \lambda mn.nm$: exp $23 \to 8$.

Пары Чёрча

Определение 8 (Пары Чёрча)

- $\text{pair} = \lambda abs.sab.$
- $\text{fst} = \lambda p.p\ \text{true}.$
- $\text{snd} = \lambda p.p\ \text{false}.$

Пример

- $\text{fst}\ (\text{pair}\ ab) \to a.$
- $\text{snd}\ (\text{pair}\ ab) \to b.$

Пара — функция, отдающая один из двух элементов по проекции.

На парах строится pred: сдвиг $(a, b) \to (b, \text{succ}\ b)$ n раз даёт $n - 1$.

Рекурсия

Проблема имён

Функции в λ-исчислении безымянны.

Факториалу нужен способ вызвать самого себя.

Пример

$\text{fact} = \lambda n.(\text{isZero } n)1(\text{mult } n(\text{fact } (\text{pred } n)))$: имя fact встречается в собственном теле, а имён в исчислении нет.

Y-комбинатор

Определение 9 (Y-комбинатор)

$$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)).$$

Свойство: $YF \overset{\beta}{\twoheadrightarrow} F(YF)$.

YF — неподвижная точка F.

Разворачиваем сколько нужно: $F(F(F(YF)))$....

Пример

Один шаг β:

$$YF = (\lambda x.F(xx))(\lambda x.F(xx)) \to F((\lambda x.F(xx))(\lambda x.F(xx))) = F(YF).$$

F — один шаг рекурсии, Y повторяет его бесконечно.

Факториал

isZero $= \lambda n.n(\lambda x.\ \text{false})\ \text{true}$.

Число n применяет функцию n раз. true уцелеет только при $n = 0$.

Пример

Fact $= \lambda f n.(\text{isZero}\ n)1(\text{mult}\ n(f(\text{pred}\ n)))$.

fact $= Y\ \text{Fact}$.

fact $2 \overset{\beta}{\twoheadrightarrow} \text{Fact}\ (Y\ \text{Fact})2 \overset{\beta}{\twoheadrightarrow} \ldots \overset{\beta}{\twoheadrightarrow} 2$.

Рекурсия без рекурсивных определений.

Рекурсия строится только из неподвижной точки.

Три комбинатора

Определение 10 (Комбинаторы S, K, I)

- $I = \lambda x.x.$
- $K = \lambda xy.x.$
- $S = \lambda xyz.xz(yz).$

Пример

- $Ix \to x.$
- $Kxy \to x.$
- $Sxyz \to xz(yz).$
- $SKKx \to Kx(Kx) \to x$, то есть SKK ведёт себя как I.

Базис S, K

S и K образуют базис: любой замкнутый λ-терм выражается через S, K и аппликацию.

Переменные и абстракция не нужны.

Абстракция скобок

Определение 11 (Перевод $[x]M$)

1. $[x]x = I$.
2. $[x]M = KM$, если x не входит свободно в M.
3. $[x](MN) = S([x]M)([x]N)$.

Пример

$[x](xy) = S([x]x)([x]y) = SI(Ky)$.

Алгоритм убирает все переменные: ни α-конверсии, ни подстановки, ни захвата.

Комбинаторная логика — ассемблер без регистров: меньше понятий, но термы громоздкие.

Вычислимость

λ-определимость

Определение 12 (λ-определимость)

$f : \mathbb{N}^k \mapsto \mathbb{N}$ λ-определима, если есть комбинатор F, такой что для всех $n_1, ..., n_k \in \mathbb{N}$:

$Fn_1...n_k \overset{\beta}{\twoheadrightarrow} f(n_1, ..., n_k)$, когда f определена.

Если f не определена на аргументе, то $Fn_1...n_k$ не имеет нормальной формы.

Неопределённость функции соответствует отсутствию нормальной формы.

Как неостанавливающаяся машина Тьюринга.

Эквивалентность

Тезис Чёрча–Тьюринга: лямбда-исчисление и машины Тьюринга определяют один класс вычислимых функций. Это тезис, а не теорема: он фиксирует границу вычислимости, его не доказывают.

Одна вычислимость, два взгляда.

- Императивные языки ближе к МТ: состояния, память, шаги.
- Функциональные — к лямбде: выражения, подстановка.

Пределы типизации

Y не типизируется в просто типизированном исчислении.

Бестиповое исчисление Тьюринг-полно, а просто типизированное — нет.

Что мы поняли?

- Вычисление — это подстановка: три конструктора — переменная, абстракция, аппликация — задают универсальный язык.
- Подстановка избегает захвата, β-редукция — шаг вычисления. Ω не имеет нормальной формы, а Чёрч–Россер гарантирует её единственность.
- Данные — это функции: булевы, числа и пары Чёрча кодируют выбор, итерацию и проекцию.
- Рекурсия — неподвижная точка Y, базис S, K устраняет переменные, а тезис Чёрча–Тьюринга отождествляет лямбду с машинами Тьюринга.

Вопросы для самопроверки

- Определите три конструктора λ-термов.
- Что такое свободная и связанная переменная?
- Зачем нужна α-конверсия и что такое захват переменной?
- Что делает β-редукция?
- Сформулируйте теорему Чёрча–Россера.
- Закодируйте and через булевы Чёрча.
- Что такое пары Чёрча и зачем они нужны для pred?
- Запишите Y-комбинатор.
- Почему YF даёт рекурсию?
- Выразите произвольный замкнутый λ-терм через S и K.